

Анализ методов и моделей облачных систем обработки данных в условиях роста объема данных и вычислительных операций в современных информационных системах

Оганесян Артем Владимирович

Независимый эксперт

Член ассоциации ИТ Сервис-менеджмент форум (itSMF России)

artem.oganesyan@gmail.com

Москва, Россия

Аннотация: Целью статьи является изучение и исследование принципа облачных вычислений и текущих основных проблем безопасности, а также их изложение простыми словами вместо сложных технических терминов. В работе были исследованы облачные технологии, без которых современный мир ИТ невозможно представить, их классификация, модели и предоставляемые им услуги.

Ключевые слова: облачная система, модели, облачные вычисления, информационные системы

Analysis of methods and models of cloud data processing systems in the context of the growth of data volume and computing operations in modern information systems.

Abstract: The purpose of the article is to study and investigate the principle of cloud computing and the current main security problems, as well as their presentation in simple words instead of complex technical terms. The paper investigated cloud technologies, without which it is impossible to imagine the modern IT world, their classification, models and services provided by them.

Keywords: cloud system, models, cloud computing, information systems

Основные понятия облачных систем обработки данных

Информационные технологии всегда считались одной из основных проблем корпоративных организаций с точки зрения затрат и управления. Однако в отрасли информационных технологий за последнее десятилетие произошел существенный сдвиг - такие факторы, как коммерциализация аппаратного обеспечения, программное обеспечение с открытым исходным кодом, виртуализация, глобализация рабочей силы и гибкие ИТ-процессы, способствовали разработке новых технологий и бизнес-моделей.

Облачные вычисления в настоящее время предоставляют организациям больше возможностей для управления инфраструктурой, экономии затрат и передачи обязательств сторонним поставщикам. Он стал неотъемлемой частью технологий и бизнес-моделей и заставил предприятия адаптироваться

к новым технологическим стратегиям [4].

Особенности облачных вычислений:

– Облачные вычисления - это новая компьютерная парадигма.
– Ресурсы инфраструктуры (аппаратное обеспечение, хранилище и системное программное обеспечение) и приложения предоставляются в виде X-as-a-Service. Когда эти услуги предлагаются независимым поставщиком или внешним клиентам, облачные вычисления основываются на бизнес-моделях с оплатой за использование.

– Основными функциями облаков являются виртуализация и динамическая масштабируемость по требованию.

– Utility computing и SaaS предоставляются интегрированным образом, даже если служебные вычисления могут использоваться отдельно.

– Облачные сервисы используются либо через веб-браузер, либо через определенный API.

NIST (Национальный институт стандартов и технологий) является признанным учреждением во всем мире для их работы в области информационных технологий. NIST определяет архитектуру облачных вычислений, описывая пять основных характеристик, три модели облачных сервисов и четыре модели облачного развертывания [5].

Есть 5 основных характеристик облачных вычислений, которые объясняют их взаимосвязь и отличие от традиционных вычислений.

1. Самообслуживание по требованию.

Потребитель может предоставлять или отменять предоставление услуг, когда это необходимо, без участия человека с поставщиком услуг.

2. Широкий доступ к сети.

Он имеет возможности по сети и доступен через стандартный механизм.

3. Объединение ресурсов.

Вычислительные ресурсы провайдера объединяются для обслуживания нескольких потребителей, которые используют мультитенантную модель с динамическим назначением различных физических и виртуальных ресурсов в зависимости от потребительского спроса [5].

4. Быстрая эластичность.

Услуги могут быть быстро и эластично предоставлены.

5. Измеренная служба.

Системы облачных вычислений автоматически контролируют и оптимизируют использование ресурсов, предоставляя возможность измерения для типа услуг (например, хранение, обработка, пропускная способность или активные учетные записи пользователей).

Существует 3 модели облачных сервисов, и эти 3 фундаментальные классификации часто называют «моделью SPI», то есть программным обеспечением, платформой или инфраструктурой как услугой [6].

– **SaaS** – программное обеспечение как сервис.

Это возможность, с помощью которой потребитель может использовать приложения провайдера, работающие в облаке.

– **PaaS** – платформа как сервис.

В этом типе услуг потребитель может развернуть, облачные или созданные пользователем приложения, созданные с использованием языков программирования или инструментов, предоставленных провайдером, в облачной инфраструктуре.

– **IaaS** – инфраструктура как сервис.

Это возможность, предоставляемая потребителю, с помощью которой он может предоставлять обработку, хранение, сети и другие основные вычислительные ресурсы, где потребители могут развертывать и запускать программное обеспечение [1].

Модели облачного развертывания

– Публичное облако - Облачная инфраструктура доступна для широкой публики.

– Частное облако - Тип облака, который доступен только для одной организации.

– Облако сообщества - В этом типе модели облачного развертывания инфраструктура облака совместно используется несколькими организациями и поддерживает определенное сообщество с общими проблемами.

– Гибридное облако - Это облачная инфраструктура, состоящая из двух или более облаков, то есть частных, общественных или общедоступных [7].

Сравнение облачных и Grid – вычислений.

Всегда были споры об эволюции облачных вычислений, и самым важным моментом в этом является Grid вычисления. Некоторые люди называют Облачные и Grid вычисления одним и тем же явлением, в то время как другие называют облачные вычисления расширением Grid-вычислений.

Grid вычисления - сложное явление, которое развилось благодаря более ранним параллельным, распределенным и высокопроизводительным вычислениям. Одно из наиболее цитируемых определений Grid-вычислений в начале было от Фостера и Кессельмана: «Grid вычисления - это аппаратная и программная инфраструктура, которая обеспечивает надежный, согласованный, всеобъемлющий и недорогой доступ к высокопроизводительным вычислительным возможностям [7]. Настоящая и конкретная проблема, которая лежит в основе концепции Grid, заключается в скоординированном распределении ресурсов и решении проблем в динамичных многоинституциональных виртуальных организациях. Обмен, который нас интересует, - это в первую очередь не обмен файлами, а прямой доступ к компьютерам, программному обеспечению, данным и другим ресурсам, как того требует ряд совместных стратегий решения проблем и посредничества в ресурсах, появляющихся в промышленности, науке и технике».

Виртуальные организации в этом определении могут быть определены как динамическая группа людей, групп или организаций, которые определяют условия и правила для совместного использования ресурсов. Некоторые организации также определили Grid-вычисления в отношении их функций.

По данным IBM: «Grid-вычисления позволяют объединять пулы серверов, систем хранения и сетей в одну большую систему, чтобы вы могли

передавать ресурсы нескольких систем одной точке пользователя для определенной цели. Для пользователя, файла данных или приложения система представляется единой огромной виртуальной вычислительной системой».

Описание облачных вычислений ранее и грид-вычислений здесь показывает, что облачные вычисления и грид-вычисления имеют много общего. Это приводит к обсуждению различий в этих двух технологиях. В приведенной ниже таблице показаны технические различия между облачными и Grid вычислениями [1,6].

	Grid вычисления	Облачные вычисления
Средства утилизации	Выделение нескольких серверов на одну задачу или работу	Виртуализация серверов; один сервер для вычисления нескольких задач одновременно
Типичная модель использования	Как правило, используются для выполнения задания, т.е. выполнения программы в течение ограниченного времени	Чаще используется для поддержки долгосрочных сервисов
Уровень абстракции	Обеспечивает высокий уровень детализации	Обеспечивает высокий уровень абстракции

Как показано в таблице, чем облачные вычисления отличаются от Grid-вычислений, так это «виртуализацией». Облачные вычисления используют виртуализацию для максимизации вычислительной мощности. Виртуализация, отделяя логическое от физического, решает некоторые проблемы, с которыми сталкиваются Grid-вычисления. В то время как Grid-вычисления достигают высокой степени использования за счет распределения нескольких серверов для одной задачи или задания, виртуализация серверов в облачных вычислениях обеспечивает высокую степень использования, позволяя одному серверу одновременно выполнять несколько задач [22].

Наряду с различиями в технологиях между Grid-вычислениями и облачных вычислений, модели использования также различаются между ними. Сетка обычно используется для выполнения заданий, в то время как



облака чаще используются для поддержки долгосрочных служб.

Как упоминалось выше, в мире технологий ведутся дебаты о том, что облачные вычисления развились из Grid-вычислений и что Grid-вычисления является основой для облачных вычислений. Таким образом, мы можем подвести итог, что Grid-вычисления является отправной точкой и основой для облачных вычислений. Облачные вычисления, по сути, представляют собой растущую тенденцию к внешнему развертыванию ИТ-ресурсов, таких как вычислительная мощность, хранилище или бизнес-приложения, и их получение в качестве услуг [11].

Внедрение облачных систем в организации, на примере *IaaS*

Хранилище *IaaS*. Услуги хранения позволяют компаниям хранить данные на устройствах хранения сторонних провайдеров. Облачное хранилище доступно онлайн и представляется клиентам в виде набора пулов или сегментов памяти, доступных с помощью богатых интерфейсов, таких как API-интерфейсы программирования, веб-интерфейсы или инструменты командной строки. Сложность архитектуры облачного хранилища скрыта от клиентов, но на внутреннем уровне она довольно сложна - обычно состоит из распределенных устройств хранения, которые управляются централизованным программным обеспечением. Сложное программное обеспечение для управления хранением использует алгоритмы, которые управляют данными, распределенными по нескольким устройствам хранения.

Однако облачное хранилище может не соответствовать потребностям каждой организации. Потенциальные недостатки включают задержку в сети, зависимость от доступности Интернета, соображения безопасности и ограниченный контроль. Задержка в сети выше по сравнению с внутренним хранилищем, поскольку центр обработки данных облачного провайдера расположен в другом географическом месте, по крайней мере, в нескольких сетевых скачках от местоположения клиента. Клиент, который хранит все данные в публичном облаке и не имеет локальной копии, полностью зависит от подключения к Интернету. Облачный провайдер должен обеспечивать высокий уровень безопасности, чтобы избежать потери или компрометации информации, а передача данных должна быть зашифрована.

Характеристики типичного облачного хранилища:

- Высокая надежность и избыточность
- Автоматически масштабируемый
- Предоставление самообслуживания для клиентов
- Доступ через богатые интерфейсы (веб-консоль, API, CLI)
- Платежная модель с оплатой по факту

Цены на облачное хранилище. Большинство провайдеров облачных хранилищ определяют цены, используя следующие факторы:

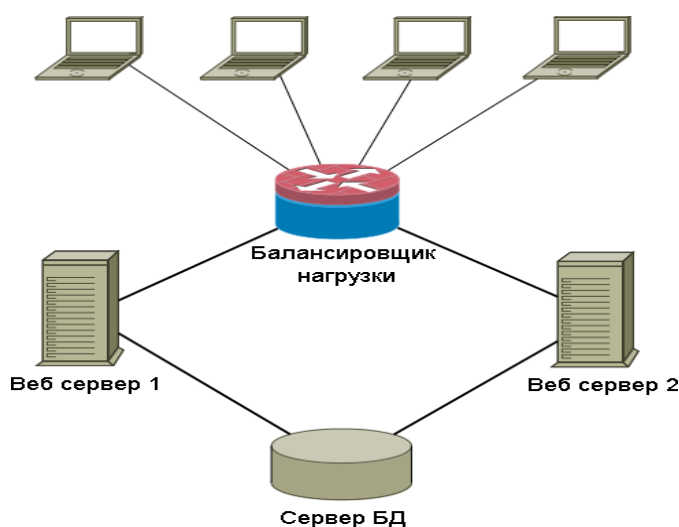
- Общее количество хранимых данных
- Количество данных, переданных в облачное хранилище
- Количество данных, переданных из облачного хранилища
- Количество конкретных http запросов

Сеть *IaaS*. Существует два основных сетевых сервиса, предлагаемых

провайдерами общедоступного облака: Балансировка нагрузки и DNS (domain name systems).

Балансировка нагрузки обеспечивает единую точку доступа для нескольких серверов, которые работают за ней. Балансировщик нагрузки - это сетевое устройство, которое распределяет сетевой трафик между серверами, используя специальные алгоритмы балансировки нагрузки [9]. Существует множество различных алгоритмов распределения нагрузки, хотя наиболее популярными являются следующие:

1. Циклический перебор: равномерное распределение соединения по всем серверам
2. Взвешенный циклический перебор: распределение соединения пропорционально весу, назначенному для каждого сервера
3. Динамический циклический перебор: аналогично взвешенному циклическому перебору, но вес сервера определяется динамически на основе непрерывного мониторинга сервера
4. Минимальное соединение: соединение отправляется на сервер с наименьшим количеством текущих соединений
5. Быстрота: распределяет новые подключения к серверу на основе самого быстрого времени ответа сервера



Балансировка нагрузки имеет несколько преимуществ: обработка отказа - в случае сбоя конкретного сервера балансировщик нагрузки автоматически перенаправляет сетевой трафик на другие серверы; производительность - поскольку нагрузка трафика распределяется между несколькими серверами, время отклика сети обычно быстрее; масштабируемость - клиенты могут быстро добавлять серверы под балансировщик нагрузки для увеличения вычислительной мощности без влияния на другие компоненты сети / системы.

Облачные сервисы с балансировкой нагрузки обеспечивают распределение нагрузки только для внутренних экземпляров, развернутых в облаке провайдера, причем цена услуги определяется часами обслуживания плюс объем используемой полосы пропускания [15]. Различные поставщики

услуг балансировки нагрузки могут предлагать разные технические функции.

Ниже приведена таблица, где мы сравниваем балансировщики нагрузки Amazon и Rackspace.

Свойство	Amazon	Rackspace
Публичный IP-адрес	доступен	доступен
Список контроля доступа	Не доступен	доступен
Регистрация соединения	Не доступен	доступен
Терминации SSL	доступен	Не доступен
Дросселирование соединения	Не доступен	доступен
LB алгоритмы	только циклический	циклический, случайный, взвешенный rr, наименьший conn.
Внутренние виртуальные IP-адреса	Не доступен (только в VPC)	доступен
Поддержка IPv6	доступен	доступен

Основные поставщики услуг облачных вычислений

Мир облачных вычислений постоянно развивается, и новые концепции и технологии создаются быстрыми темпами. Каждый из существующих и новых поставщиков услуг предлагает различные облачные решения, предназначенные для разных категорий клиентов, т.е. приложения SaaS для отдельных пользователей или услуги IaaS для предприятий. Разнообразие создает конкуренцию. Множество доступных облачных решений и поставщиков облачных услуг ускоряет скорость, с которой сервисы облачных вычислений развиваются в функциональном отношении. Кроме того, усиление конкуренции приводит к удешевлению услуг, что приводит к увеличению выгод для клиентов. Несмотря на то, что они ориентированы на разные типы клиентов и используют разные технологии, у поставщиков облачных услуг есть одна общая цель: предлагать стабильные, безопасные и масштабируемые облачные вычисления, которые облегчают разработку приложений и снижают затраты на управление локальной средой. Примерами основных поставщиков услуг облачных вычислений являются Amazon, Google(GAE- Google App Engine) и Microsoft (MWA-прикладная платформа Microsoft, доступная для широкой публики, которая предоставляет ресурсы в общедоступной облачной инфраструктуре) [19].

В таблице 1 представлено сравнение между тремя платформами облачных сервисов, рассмотренными выше. Данные, представленные в

таблице, взяты из официальной документации, предоставленной тремя поставщиками облачных услуг, по состоянию на апрель 2020 года. Сравнение проводится с технической и функциональной точек зрения.

Таблица 1. Сравнение между различными платформами облачных сервисов

Тип развертывания	Amazon EC2	MWA	GAE
	IaaS	PaaS/IaaS	PaaS
Поддерживаемые ОС	Red Hat Enterprise Linux, SUSE Linux Enterprise, Fedora, Oracle Enterprise Linux, various Linux distributions, Windows Server, Amazon Linux AMI	Windows Server, CentOS, Ubuntu, SUSE Linux Enterprise, openSUSE, GuestOS	Linux
Поддерживаемые языки	.NET, C++, PHP, Ruby, Python, Java, Android, iOS	.NET, C++, PHP, Ruby, Python, Java, Windows Phone, iOS	Java, Python, Go, PHP
Хранилище	MySQL, MongoDB, Microsoft SQL Server, PostgreSQL, CouchDB + others	Azure SQL, нереляционные таблицы, очереди, blobs	Google Cloud SQL, Cloud Storage, App Engine Datastore
Приложения запускаются в виртуальных машинах	Да	Да	Нет
Интерфейс управления	API, Command Line, User Interface (UI)	API, Command Line, Management Portal	API
Балансировщик нагрузки	Да	Да	Да
Подходит для развития малых размеров приложений	Да	Нет	Да
Подходит для развития больших размеров приложений	Да	Да	Да
Средства мониторинга	CloudWatch	Интернет-портал управления, 3 rd Party	Appstats for Java, Appstats for Python
Модель ценообразования	почасовой, планы подписки	почасовой, планы подписки	на основе использования
Бесплатные ресурсы	30 Гб для хранения, 15 GB исходящего трафика ежемесячно, в течение одного года	3-месячная пробная версия с доступом ко всем функциям, но с ограниченными ресурсами	1 Гб для хранения, 1 Гб исходящего трафика ежемесячно

Также здесь рассмотрим, **Microsoft Azure** как облачную платформу

Microsoft Azure (ранее известный как Windows Azure) - это общедоступная облачная платформа, созданная и размещаемая Microsoft. Он был выпущен 1 февраля 2010 года и с самого начала должен был конкурировать на относительно молодом рынке, где доминируют Amazon Web Services. За прошедшие годы Azure повзрослела, и по состоянию на май 2014 года она стала одним из лидеров в облачных вычислениях, все еще далекая от достижения Amazon, но уже опережая другие конкуренты. Одним из факторов, способствующих такому прогрессу, была политика относительной технологической открытости Microsoft. Компания не ограничивала доступную операционную систему семейством Windows, но поставляла машины с различными дистрибутивами Linux, часто с предустановленным популярным программным обеспечением. Более того, разработчики не ограничиваются поддерживаемыми Microsoft языками, такими как C #, Visual Basic или C ++, но и другими популярными, например, Java, PHP, Python или Ruby. Управление учетной записью Azure возможно через веб-приложение под названием Microsoft Azure Management Portal или через различные API-интерфейсы.

Модели выполнения. Модели выполнения - это способы, с помощью которых выполнение приложений обрабатывается Azure. Они отличаются простотой использования, универсальностью, свободой управления и предполагаемым типом приложений. В Microsoft Azure доступно четыре модели выполнения:

- Облачные сервисы
- виртуальные машины
- веб-сайты
- Мобильные услуги

Облачные сервисы. Облачный сервис обеспечивает относительно простой способ размещения приложения в облаке. Это решение PaaS, поэтому оно имеет удобные функции, но в то же время предъявляет ряд требований. Многие задачи автоматизированы или упрощены, например, автоматическая установка системных исправлений и выкатывание новых версий образов. Кроме того, вам не нужно беспокоиться о конфигурации системы - вам нужно только предоставить пакет с вашим приложением и конфигурацией. Конфигурация должна, среди прочего, содержать список «ролей», используемых приложениями. Проще говоря, они являются предварительно сконфигурированными виртуальными машинами и существуют в двух вариантах:

- 1) Веб-роль - предназначена для размещения веб-приложений, с установленным IIS
- 2) Рабочая роль - лучше всего подходит для серверных услуг

Уровень облачной службы представляет ресурсы, которые обеспечивают вычислительную мощность для приложения, развернутого в среде Windows Azure. Сам облачный сервис представляет собой сервис, который обеспечивает вычислительную мощность и содержит фактический код приложения. В MWA приложения делятся на три компонента, называемые ролями, а именно веб-

ролями, рабочими ролями или ролями виртуальных машин. Веб-роли представляют собой внешний интерфейс приложения, скоторым пользователи могут взаимодействовать, тогда как рабочие роли - это внутренний компонент, часть приложения, которая работает в фоновом режиме и не видна с точки зрения пользователей. Любое приложение MWA должно быть реализовано как одна роль, или комбинация ролей, и одновременно могут выполняться несколько экземпляров любой роли приложения. Кроме того, веб-роли и рабочие роли функционируют независимо, и каждая роль выполняется в своей отдельной виртуальной машине. Несмотря на то, что они работают в разных виртуальных машинах, разные роли могут взаимодействовать друг с другом, но пользователи приложений не могут напрямую взаимодействовать с рабочей ролью.

Услуги хранения. Microsoft Azure предоставляет способ хранения данных, которые могут совместно использоваться развернутыми приложениями, виртуальными машинами или доступны извне, которые обычно используются или создаются развернутыми приложениями. С пользователя взимается плата за количество хранимых данных, выбранные параметры репликации, количество запросов на чтение и запись и передачу данных за пределы региона службы хранения. Существует 4 так называемых службы хранения:

1. Блоб хранилище
2. Хранилище очереди
3. Таблица хранения
4. Файловое хранилище

Блоб хранилище предназначен для хранения больших объемов неструктурированных двоичных данных. Это делает его наиболее универсальной службой хранения, доступной в Windows Azure. Данные хранятся в виде капель. Есть два типа капель:

1. Block blob оптимизирован для потоковой передачи и хранения облачных объектов. Это хорошо для эффективной загрузки больших объемов данных.
2. Page blob - страничный блоб оптимизирован для произвольного доступа. Это коллекция из 512-байтовых страниц. Страницы BLOB-объектов обычно используются для хранения образов жесткого диска.

Хранилище очереди. Хранилище очередей предоставляет решение для обмена сообщениями, обычно используемое для связи между отделенными компонентами приложения. Это дает пользователю надежный интерфейс Get \ Put \ Peek. Сообщение, вставленное в очередь, может содержать до 64 КБ двоичных или текстовых данных. Практически все операции управления Azure можно выполнять через веб-портал управления Microsoft Azure, но для этого требуется, чтобы пользователь просматривал все меню и окна. Microsoft предлагает несколько способов автоматизации управления Azure с помощью нескольких API:

- Собственные SDK для .NET, Java, Python, Ruby, Node.js, PHP и мобильных платформ (iOS, Android, Windows Phone)
- Командлеты PowerShell
- Кроссплатформенный интерфейс командной строки

• REST API

База данных Azure SQL. База данных Azure SQL - это облачная реляционная база данных, разработанная и поддерживаемая Microsoft. Основанный на технологии SQL Server, Azure SQL - это масштабируемая, высокодоступная и безопасная служба базы данных. В отличие от развертывания базы данных локального сервера SQL, для развертывания базы данных Azure требуется, чтобы разработчик управлял логическим сервером. Microsoft управляет и поддерживает физические ресурсы, необходимые для работы базы данных Azure. Кроме того, поскольку разработчики не имеют контроля над физическими ресурсами, они не могут выбрать физический жесткий диск, на котором хранятся файлы базы данных. Однако конечная точка, через которую можно получить доступ, остается прежней.

Базы данных Azure SQL выигрывают от плавного автоматического устранения сбоев оборудования и автоматической репликации данных. Кроме того, Azure SQL предназначен для повышения производительности и высокой масштабируемости. Разработчики могут создавать и развертывать базу данных в течение нескольких минут, а размер базы данных автоматически масштабируется при достижении лимита хранилища, если позволяет тарифный план.

Служба Azure SQL находится за сетевым брандмауэром. По умолчанию все подключения к серверу Azure SQL отключены, за одним исключением. IP-адрес, который использовался для создания сервера SQL, имеет доступ, но только через порт 1433. Разработчики должны перейти к меню конфигурации сервера баз данных Azure SQL на портале управления и указать, какие IP-адреса могут обращаться к серверу базы данных. Кроме того, брандмауэр базы данных позволяет разработчикам блокировать доступ приложений MWA к конкретным базам данных.

Еще одна функция безопасности - проверка сертификата. Базы данных Azure SQL принимают сообщения только в зашифрованном виде и отклоняют соединения от приложений с недействительными сертификатами безопасности. Кроме того, установленные соединения с Azure SQL сбрасываются каждый час, поэтому разработчики должны повторно вводить учетные данные администратора базы данных.

Azure AppFabric. Azure AppFabric является важным компонентом архитектуры MWA и предоставляет службы, которые отделяют платформу MWA от других облачных платформ. Он представляет средний уровень платформы MWA, служа мостом связи между приложениями, работающими в разных системах. Одно из преимуществ использования сервисов AppFabric заключается в том, что он упрощает процесс управления приложениями, тем самым повышая производительность разработки. Двумя основными компонентами AppFabric являются служебная шина и контроль доступа. Разработчики могут использовать службы, предоставляемые AppFabric, для безопасного соединения компонентов приложения вместе, то есть различных ролей, и простого управления доступом к ресурсам. Эти две службы необходимы для большинства облачных приложений, и они полагаются и дополняют друг

друга для достижения наилучших результатов подключения.

Выводы:

Рост облачных вычислений знаменует собой начало новой эры информационных технологий, которая вступает в модель агрегированных вычислений. Между тем, это приводит к трансформации приложений из локальной в облачную среду, что значительно улучшает нашу жизнь и работу. Более того, облачные вычисления популяризировали таинственные и неуловимые профессиональные концепции, такие как распределенные вычисления, параллельные вычисления, виртуализация и так далее. Тем не менее, по сравнению с другими терминами, название «Облако» кажется абстрактным и иллюзорным, и трудно представить связь между «Облаком» и ИТ. С другой стороны, для публики будет более приемлемым и понятным описание технологии визуально. Кроме того, эта энергично развивающаяся отрасль будет влиять на другие области в будущем. Например, облачные вычисления будут сильно стимулировать развитие интернет-индустрии, поскольку они являются основой для реализации облачных вычислений, в то время как их интенсивное влияние на аппаратную индустрию можно предсказать, поскольку облачные вычисления обработали сложные вычисления и доставили результаты клиентам.

В связи с постоянным расширением облачные технологии будут использоваться все больше и больше. Между тем, проблема, связанная с проблемами безопасности, является беспрецедентной, что требует постоянного изучения исследователей в области ИТ и информационной безопасности. Следовательно, такие известные организации, как CSA, поддерживающие безопасность облачных вычислений, были созданы для того, чтобы способствовать наилучшему использованию облачных вычислений и обеспечить надежную защиту для них.

В целом, облачные вычисления - это новая модель, основанная на расширении, использовании и взаимодействии Интернета. В статье подробно рассмотрены концепция, структура, развертывание, типичное использование и основные проблемы безопасности облачных вычислений.

Список литературы

1. Грегг Б. «Производительность систем: Enterprise и Cloud», 2014.
2. Мурзин Ф.А. «облачные технологии: основные модели, приложения, концепции и тенденции развития», 2016
3. Миронович В. «Обзор: Облачные вычисления», 2017
4. Томас Эрл, Рикардо Путтини, Зайгам Махмуд, «Облачные вычисления: концепции, технологии и архитектура», 2018
5. Кис Блокленд, Йерун Менгеринк, Мартин Пол, «Тестирование облачных сервисов: как тестировать SaaS, PaaS и IaaS», 2017
6. Сангита Н. «Облачные вычисления и технологии виртуализации в библиотеках», 2017
7. <http://www.bureausolomatina.ru> статья «Будущее облачных технологий: европейский взгляд»

